

UNCLASSIFIED

Defense Technical Information Center Compilation Part Notice

ADP010698

TITLE: Direct Manipulating Interface Techniques
for Users Interacting with Software Agents

DISTRIBUTION: Approved for public release, distribution unlimited

This paper is part of the following report:

TITLE: Usability of Information in Battle
Management Operations [l'Exploitation de
l'information dans les operations de gestion du
champ de bataille]

To order the complete compilation report, use: ADA389629

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, ect. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:

ADP010683 thru ADP010703

UNCLASSIFIED

DIRECT MANIPULATION INTERFACE TECHNIQUES FOR USERS INTERACTING WITH SOFTWARE AGENTS

Michael Young

&

Robert G. Eggleston

Air Force Research Laboratory

Human Effectiveness Directorate

2698 G Street

Wright-Patterson Air Force Base, OH 45433

USA

&

Randall Whitaker

Logicon Technical Services, Inc.

PO Box 317258

Dayton OH 45437-7258

Summary

In this paper we provide an overview of our research investigating what functionality should be provided to users of a future *Joint Battlespace Infosphere (JBI)*. We characterize and discuss the development of the JBI as a new form of automation that employs intelligent agents to autonomously seek, retrieve, and fuse information. We believe that the development of new types of direct manipulation interfaces are the best approach to achieving JBI goals of reducing decision time and manning while maintaining positive control over the command and control (C2) system. Further, we argue that the integration of direct manipulation interface techniques with interface agents will change the HCI from a mechanism to execute tasks into a decision-aid that supports cognitive information processing. We contextualize this discussion by providing an overview of the Air Force Research Laboratory's Human Interaction with Software Agents (HISA) project. This effort is developing a HCI for Air Mobility Command's (AMC) Tanker Airlift Control Center (TACC) that interacts with operational C2 systems through intelligent agents, similar to the manner of the proposed JBI.

I. The Joint Battlespace Infosphere Concept (JBI) as a Three-Layer Model

To achieve the goal of information dominance on the battlefield, the U.S. Air Force is exploring the development of a *Joint Battlespace Infosphere (JBI)*. The JBI is proposed as an integrated C2 system encompassing the entirety of Air Force operations. The JBI will be

implemented through data and communications networks which will enable warfighters to plug into the JBI in a fashion analogous to logging onto the Internet. The JBI will then provide these warfighters access to the complete range of information products and services necessary for operational decision making. As a comprehensive decision making environment, the JBI would serve as both the repository and generator of mission critical data. Individual warfighters would provide data for the JBI and in turn receive fused presentations of information tailored to their goals and needs.

Two of the primary goals of the JBI concept are the reduction of warfighter decision times and staffing demands. The first goal is to be obtained by more efficiently accessing and fusing decision-critical data and more effectively presenting it for employment in decision making tasks. The second goal is to be obtained by eliminating inefficiencies and redundancies in the current 'stove-piped' information architectures among numerous operational units. Both goals can be obtained to the extent the cognitive burdens of decision makers are reduced. Reducing decision makers' cognitive burdens serves the first goal by facilitating the decision processes of any given decision maker. It serves the second goal by leveraging decision making efficiency to permit smaller staffs to equal or exceed current performance standards.

Implementation of the JBI requires translating the concept into deployable decision support tools and systems. Our analysis of the JBI concept suggests that the requisite network-oriented deployable products will evidence three types of functionality, which in turn can be described in terms of three layers:

- A *network services layer* which provides connectivity between the various C2 systems;
- an *application services layer (ASL)* which provides services such as planning, scheduling, and information fusion, often mediated through intelligent agents; and
- a *human computer interface (HCI)* layer through which warfighters receive information and enact operational tasks.

II. Agent Support Criteria for the HCI and ASL Layers

The primary areas for intelligent agent support will be the ASL and HCI layers, and it will be these two layers upon which our discussion will focus. It is important to clearly distinguish between these two layers, primarily because their respective implementations will employ intelligent agents, albeit quite differently (Milewski & Lewis, 1997). Although both involve multiple processes (i.e., agents) communicating with one another in an intelligent fashion, our project experience suggests it is critical to 'frame' the purpose of these layers in distinct ways. This 'framing' allows for more precise identification of agents' roles, as well as illuminating the optimum referential context for design and implementation of agent support, for each layer.

To illustrate, in the following paragraphs we shall summarize the ASL and HCI layers with respect to three critical dimensions of agent implementation:

- *ontology* – the fundamental semantics underlying terms of reference and types of inference.
- *homo-/heterogeneity* – the differential unity / multiplicity of elements (or element types) engaged by users, the agents, or both.
- *autonomy* – the degree to which a given agent (or class of agents) functions outside the scope of user monitoring and/or direction.

II.A. The ASL Layer

The ASL is best characterized as a functional architecture designed to accomplish specific tasks such as scheduling aircraft and crews for specific missions or planning the movement of forces into a theater of operations. Given the diversity in both legacy and prospective mission-critical systems, the ASL must provide for a flexible mapping of tasks to computers (or computer processes). In a traditional C2 system a task (e.g., scheduling) is often accomplished by a specific computer running a specific (scheduling) application program. In contrast, in the JBI tasks will not be mapped to specific hardware / software platforms. Agents working for a specific user will request a service, and other agents will manage the details of how and where that service is actually

accomplished (e.g., on which computer; using which software). This illustrates two important points. First, the 'ontology' for ASL agents must emphasize procedural logic, support systems, data routing protocols, etc. Second, the multiplicity of items (and item types) referenced in this ASL ontology means that heterogeneity of functionalities (and loci of functionalities) will be a key concern in ASL design and implementation.

This heterogeneity extends to the agents themselves. That is, the ASL layer will not exhibit a single standard language, type of agent, or form of agent communication. Instead, there will be a variety of agents that speak a diversity of languages (e.g., Knowledge Query Mark-up Language [KQML] {Finn, Labrou, & Mayfield, 1997}, Knowledgeable Agent-oriented System [KaoS]{Bradshaw, Deutfield, Benoit, & Wolley, 1997}) and employ a number of communication protocols. This conglomerate of distributed disparate agents will advertise and broker services among themselves in order to find the optimal means to complete a specific task in the operational context of priorities, situational constraints, and other related tasks underway.

Agents' 'brokerage' of diverse goals, tasks, conditions, and functionalities will add value to the extent that it manages the relevant complexity (i.e., complexity of type – heterogeneity) on behalf of (e.g.) planners and commanders. In this case, the obvious tactic for complexity management is to allow the agents to automatically handle the details of user-defined tasks. Phrased another way, ASL agents' value will be directly proportional to the amount of detailed tasking they can accomplish without users' direct inspection and guidance. As such, the hallmark of useful ASL agents is capacity for autonomous action (vis a vis the user).

II.B. The HCI Layer

In contrast, the HCI layer is defined with respect to the user him/herself. In a traditional C2 system, most tasks are initiated and managed by a user. In contrast, design goals for the JBI include reducing the decision-making cycle time, while simultaneously reducing the number of personnel, and while maintaining positive (human) control over the weapon systems. The HCI layer, then, must provide the capacity for user inspection of task parameters as well as the means through which the user invokes and manages his/her tasks. In contrast to the ASL layer, the HCI layer is best characterized as an architecture of forms (as opposed to functions) designed to facilitate understanding of and control over specific tasks. More specifically, it is implemented as a collection of graphical user interface [GUI] widgets that actualize a user-interface model in which the user delegates to and collaborates with intelligent software agents.

The HCI layer is effective to the extent it facilitates non-autonomous actions – i.e., those actions reserved for direct human control. This fact underlies one critical distinction between design approaches to the ASL and HCI layers. In the ASL layer, the opportunity for agent autonomy affords designers the ability to design with respect to any functionalities the software agents can implement. In the HCI layer, the requirement for discretionary user control (i.e., HCI agent non-autonomy) forces designers to constrain themselves to prioritizing those particular functionalities the human can and/or must manage.

The most obvious thing the human must manage is his / her interactivity with the HCI layer itself. As the primary point of engagement between user and system, the HCI layer is the explicit ‘point of service’ for the JBI. For the JBI to support effective work and decision aiding, the HCI layer must itself be designed as an effective work / decision aid. This means that the HCI layer must be designed so as to reflect key referential and operational aspects of the task and related decision space(s). As such, the HCI layer must be based on an ontology consistent with the user’s viewpoint – i.e., an ontology focusing upon the mission, specific tasks, and decisions.

The kind of diversity (heterogeneity) that is usefully exploitable in the ASL layer is itself a serious problem for humans to routinely handle. The user’s cognitive workload should not be increased by forcing him/her to deal with details of the HCI layer’s implementation –

e.g., the specifics of how the HCI agents interact with the ASL agents. This means the HCI layer must emphasize both simplicity (non-complexity) and consistency of both form and function in portraying and addressing tasks, conditions, tools, and functionalities. As such, homogeneity must rule in any HCI. That is, a user must be able to rely upon a GUI widget coherently displaying task parameters and consistently performing any functions he/she invokes in response. If a widget displays the status of an airbase in one fashion at one time, the user should expect it to portray that status in the same fashion (a) for other airbases anytime and/or (b) that same airbase some other time. Similarly, if a widget allows specific actions (e.g., drilldown to more detailed status data) on one occasion, the same functionality should be predictable the next time the widget is used.

II.C. Summary of ASL / HCI Design Tradeoffs

Figure 1 is offered as a summary illustration of these points. The ‘squares’ represent interface elements visible to the user, and the ‘spheres’ represent the software agents servicing the interface as well as accomplishing the ASL layer functions.

At the interface, the HCI layer offers specific functionality (e.g., scheduling missions, requesting resupply, etc.). The user may not know (or care) that the functionality being provided is mediated through interface agents (the spheres clustered behind each interface element). He/she may not know because the interface reflects a task-oriented (as opposed to a system-

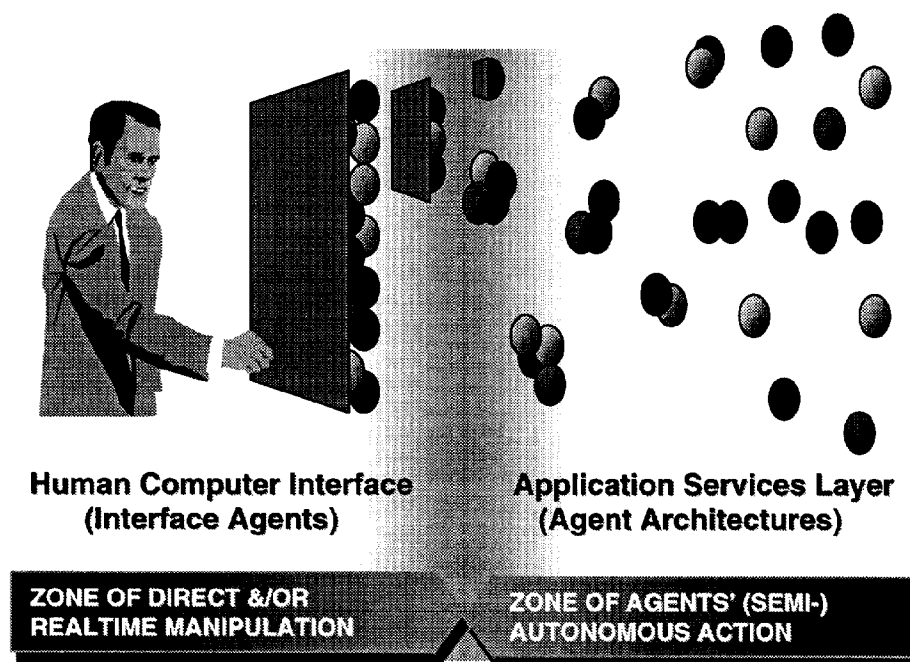


Figure 1: The HCI and ASL layers

oriented) ontology. He/she may not be aware that as we move farther into the ASL layer, agents increasingly interact autonomously among themselves, brokering services and accomplishing tasks. Conversely, as we move in the other direction (from ASL toward HCI), agents (and/or agent sets) increasingly reflect task-oriented factors, and may be provided interface elements by which the user can address them directly (e.g., setting functional preferences).

In between the extremes of the ASL and HCI deployment styles is the gray area depicted in Figure 1. This is the domain where the homogenous HCI layer's agents interact with the heterogeneous agent-based ASL architectures. It is also the point at which full agent autonomy comes into play. Because it represents the extent of HCI service 'coverage', this middle ground demarcates the user's zone of awareness with respect to the JBI. On the one hand, it is from this point rightward (cf. Figure 1) that autonomous agents can be trusted to operate out of sight of the user. On the other hand, it is from this point leftward that agents are usefully made 'visible' for user inspection and 'available' for user manipulation. For example, users may want to "drill-down" into this area to tailor agent behavior – e.g., selecting a specific module for a particular task or creating new agents to serve as sentinels for problematical conditions. As such, it is in this middle ground that tradeoffs must be determined involving our two critical factors of homo-/heterogeneity and autonomy.

Perhaps even more importantly, this middle ground is the point at which the ASL and HCI layers' ontologies must intersect and interoperate. With respect to Figure 1, it is from this point rightward that the ASL layer's system-oriented ontology may prevail, and it is from this point leftward that the HCI layer's task-oriented ontology must prevail. For the above-cited drill down to be effective, there must be semantic interoperability between the users' conceptual work domain model (via the analogous HCI layer task ontology) and the ontologies of the ASL agents brokering services among planners, schedulers, search engines, etc.

The most difficult challenges facing JBI developers concern interoperability tradeoffs between the HCI and ASL layers as described above. They must provide for interoperability between the users' homogenous interface and the heterogeneous agent world. This functional interoperability must be qualified with respect to reasonable allocation of users' control versus agent autonomy. Finally, the distinct semantic priorities of the ASL and HCI layers must be interwoven through interoperability of their respective ontologies. These difficult goals must be achieved in a manner that (a) maximizes functionality provided the users; (b) minimizes users' cognitive workload; (c) maximizes

system operational efficiency; and (d) promotes task effectiveness in JBI applications.

III. Our Work-Centered Interface Concept

In this section we review some of the design goals we hope to achieve by creating a separate HCI layer, where the user's interaction with the system is mediated through interface agents. First, we outline what we see as the primary problem – cognitive burdens on the decision maker entailed in addressing two distinct ontologies (domains of reference and knowledge). Second, we identify two major design goals for alleviating this problem. Finally, we present the HISA design criteria through which we pursued these design goals.

III.A. The Problem: Complexity of Ontological Reference in User / System Interaction

In general, every computer-implemented decision aid can be differentiated into two distinct functional components:

- The *decision-making component* supports task-specific decision making (e.g., deconflicting an aircraft scheduling problem.). The decision-making component must be configured so as to allow the user to address the task he/she is executing.
- The *information manipulation component* supports task-specific data / information activities (e.g., accessing a system to retrieve data or to assign a new mission start time. The information manipulation component must be configured so as to allow the user to address the tools (information systems) he/she employs in the course of the task.

Owing to this dichotomy of reference, these two decision aid components differ in the types of knowledge that must be active. A decision-making task requires activation of a *task domain ontology* – i.e., the set of specialized terms, meanings and relations between terms that captures or represents the subject matter itself (i.e., the domain knowledge of scheduling goals and constraints). Information manipulation requires activation of a *system ontology* – i.e., the set of specialized terms, meanings and relations between terms that captures or represents working knowledge of the subject matter documentation.

Consequently, a user engaged in decision-making must engage in multi-tasking behavior which involves (potentially extensive) shifting between the frames of references (or activate ontologies) of the systems and task. Let us illustrate this with an example. To deconflict a scheduling clash, a mission planner may have to access two or more systems. Interacting with each system requires the planner to develop and execute

an information manipulation strategy. This may require several procedural steps such as logging on to a system, accessing the appropriate data base, and then executing a query. Phrased another way, the user must generate and work through a plan distinct from, but potentially of similar complexity as, the mission plans involved in the scheduling clash. In utilizing the retrieved data for deconfliction, the planner must then turn to an entirely separate referential framework reflecting (e.g.) aircraft, airfields, and planning constraints. In other words, the planner must invoke and refer to a referential set distinct from, but potentially of similar complexity as, the data dictionaries underlying the retrieved data.

The user must therefore grapple with developing a single problem solution via engagement with two distinct referential and procedural frameworks. This increases the user's work demands, cognitive burdens, and risk of error. With respect to our deconfliction example, the user encounters transcription costs in assembling relevant data from multiple sources (e.g., writing down or printing out conflicted missions' data as it comes in). Once all the relevant data is at hand, the user must then endure the interpretation costs for interrelating a set of data field entries and a set of mission arrivals / departures at the given airfield.

III.B. The Solution: Minimizing Ontological Complexities to Reduce User Cognitive Complexity

The above-cited costs are a matter of *cognitive complexity*. Cognitive complexity (Chechile, Eggleston, Fleischman, & Sasseville, 1989) is a measure of how much cognitive resources are required to execute an activity. Note that cognitive complexity for an activity is an aggregate of complexity of the information manipulation and decision-making components. Cognitive complexity for an information manipulation task is usually a function of how much planning is required to execute a task. Cognitive complexity for a decision-making task is harder to quantify because of the variability of the types of tasks the actor is engaged in and the role the actor's skill level plays in task performance.

This dilemma would be minimized to the extent the system and task ontologies correspond (e.g., in terminology). Unfortunately, this correspondence is rarely evident in management information systems. Moreover, the increasingly integrated network character of emerging command and control architectures are predicated upon the ability of warfighters to 'drill down' (into their own data assets) and 'reach back' (for data assets possessed by someone else). Because the trend is toward increased referential qualifications (drill down) or more numerous data sources (reachback), the above-cited

ontological dilemma can only become more problematical.

Much of the HISA interface design effort was directed toward enhancing task / system correspondence and reducing mission planners' current reliance on work-around strategies and tools (e.g., pen and paper). Our design goals for the agent-based HCI layer focused upon providing mission planners with more direct support for their decision-making through:

- increasing the time the user operates "on-task" – i.e., accomplishes task activities by working with reference to the task domain.
- reducing the amount of time the user digresses "off-task" – i.e., is sidetracked into activities requiring reference to the system domain.

III.C. Our Approach: A 'Work Centered' Interface Style

The two key solution criteria cited above must be reflected in design and development work to obtain the expected payoffs. To accomplish this, we translated the goals and principles cited above into a set of HISA interface design criteria to guide our work. These criteria reflect the following priorities:

- maximize explicit reference to task domain elements in the on-screen HISA information displays
- maximize cross-reference among HISA information displays with respect to core task domain concepts (e.g., missions, airfields, courses of action)
- minimize procedural costs for accessing and retrieving relevant data (e.g., by automating queries)
- maximize effective fusion of data from the multiple databases with which the planners must currently interact (e.g., by assembling a single airfield summary view from data scattered across numerous database tables)
- minimize cognitive burdens for identifying, seeking, and/or interpreting relevant information (e.g., by reducing interpretational demands)

The implementation strategy uniting the above-cited design goals and approaches entailed a trade-off between (a) the interfaces engaged by the users and (b) the functionalities delegated to the software agents. We strove to configure the display components to prioritize task domain referentiality, and we prioritized allocating system domain-oriented actions (e.g., database access) to the agents.

This is not simply a matter of providing a highly graphical direct manipulation user interface. Direct manipulation capabilities do help reduce the cognitive

complexity of the information manipulation component by making it easier to directly manipulate information elements. In the eventual realization of the JBI concept, warfighters must access multiple systems, each one of which may provide a very different interface. Direct manipulation, accordingly, does not necessarily eliminate the need to switch ontologies. As a result, direct manipulation alone is not sufficient to accomplish our design criteria and hence our design goals.

Interface concepts can be characterized with respect to either the perspective of the user (e.g., direct manipulation) or the perspective of the system(s) (e.g., object oriented, agent-based). We believe the key innovations of our HISA effort, though involving novel system capabilities, are best characterized from the user perspective. Though the HISA interface elements (as viewed by the user) certainly represent 'direct manipulation', this label does not convey what we see as the really innovative aspect of this work. Our HISA interface concepts direct as much of this direct manipulation as possible to task (as opposed to system) elements.

In other words, we are attempting to more directly support task decision-making by effecting a closer correspondence between on-screen display elements and elements of the task domain (as opposed to elements of the information space). In effect, we are making the system more 'transparent' vis a vis the mission planners' tasks. This strategy reduces the cognitive complexity involved in addressing task activities by reducing the procedural and interpretational overhead for addressing task issues through the 'lens' of support system-specific interfaces. This allows us to maximize the time the user spends oriented to the task domain itself by maximizing his/her ability to address task activities in terms of task (as opposed to system) ontology. We call an interface which realizes our design criteria *work centered*.

IV. Our HISA Products as Work Centered Interfaces

The Air Force Research Laboratory (AFRL) has been researching work centered interfaces as part of the "Human Interaction with Software Agents" (HISA) project. The target worksite for HISA products is the Tanker Airlift Command Center (TACC) – a mission planning and execution center within the USAF Air Mobility Command (AMC). TACC units plan, schedule, and monitor airlift missions on a continuous basis. More particularly, HISA has concentrated on the specific category of *channel missions* – those missions which are routinely conducted along established routes. Key characteristics of the USAF channel mission planning work include:

- *Long lead times for mission plans.* Channel mission plans are typically drafted and published months

ahead of time (in advance of mission take-off) to enable organizations to plan family moves.

- *Heterogeneous data assets.* Missions are initially planned using one system, with the final version being 'published' to another. These systems differ in the way the data tables are laid out. Further, multiple other databases each uniquely contain relevant information such as (e.g.) airfield restrictions and alerts on airfield status.
- *High cognitive burden for data access.* The multiplicity and diversity of record tables make it laborious to track down specific details of a mission. When obtaining such details require access to multiple tables and/or other databases, planners must execute multiple queries – potentially involving multiple search syntaxes.
- *No capacity for unified issue visualization.* The scheduling system provides only structured textual records of mission data, arranged by mission. To review issues involving multiple missions, planners must often execute a query, print out the results, and review this printout manually. Discerning on-ground conflicts at a given airfield typically requires interrelating mission text entries by drawing lines among them with a pen.
- *Little or no automated decision support.* The system provides no automated inference to detect conflicts among mission plans as they are accreted. Moreover, the system provides no automated alerts on conflicts and other problematical conditions.
- *Discontinuous situation awareness.* Once a channel mission is published (months ahead of time), conflicts resulting from subsequently-published missions can go undetected (and hence unresolved) until it is nearly time to launch the mission.
- *High potential for time-critical problem solving under duress.* In accordance with TACC business rules and policies, channel planners must usually defer to planners of other missions types (e.g., contingency missions) when resources (e.g., aircraft) are insufficient to execute all plans at once. The above-cited conditions make for frequent last-minute replanning problems, while the channel missions' low prioritization diminishes planners' ability to definitively resolve those problems on their sole initiative.

As of the time of this writing, the HISA effort had produced design specifications for a work-oriented planner interface, as well as dynamic demonstration models for some core elements of this interface. Our HISA interface elements have been demonstrated in real-time interoperability with networked data sources, providing concise and coherent displays of mission planning parameters as well as offering proactive support (e.g., alerts; plan conflict data) for decision makers. The following sections offer selected examples of our HISA interface elements and illustrate how they both (a)

address problems faced by the client TACC channel mission planners and (b) illustrate the principles, goals, and criteria outlined earlier in this paper.

IV.A. The Foundation for Work-Oriented Interface Design: A Task Ontology

Agent-based support will afford us the ability to shift the users' 'field of vision' from the machine to the task itself. The 'intelligence' of ASL and HCI agents will relieve users of cognitive burdens attributable to having to understand the mechanics of the support system to get a task accomplished. As a result, an agent-based HCI layer allows an unprecedented ability to reflect the ontology of the task rather than the ontology of the system(s). By disengaging the task semantics from the tool semantics, we have been able to design our HISA HCI layer elements to directly reflect the mission parameters comprising the critical issues in the planning process, as opposed to the planning artifacts (e.g., cryptic database records) reflecting the limitations of the planning documentation (Eggleston, 1993).

The first step in accomplishing this required the development of a coherent task ontology which was consistent with the key referential, inferential, and procedural elements by which users engage their work. It was obvious from the start that the primary object of task engagement was the mission plan – e.g., the documented record of a scheduled mission as stored in GDSS. However, it was equally obvious that the problematical issues listed above all related to grappling with this mission plan documentation at the expense of efficiently and effectively addressing the subject matter documented. Our first goal was to identify the key subject matter on the way to configuring the HISA interfaces to reflect it.

The initial knowledge acquisition efforts clearly indicated the primary object of referential and inferential engagement was the mission itself – i.e., the act of employing an aircraft and crew to transport a specific set of items from one airfield to another. We therefore nominated "mission" to be the core construct around which to develop the mission planner task ontology. Further analysis (e.g., of actual and representative problem scenarios) resulted in our subdividing this core construct into three components:

- *Port* – Either one of the airfields involved in a given mission leg.
- *En Route* – The passage of the loaded aircraft from one Port to the other.

- *Package* – The aircraft, crew, load, and other items required to perform a mission leg.

Our early knowledge acquisition indicated that problems were typically delimited with respect to one or another of these components. For example, lack of a functional aircraft was a Package issue. Similarly, weather-motivated diversion to an alternate landing site was an En Route issue, and exceeding the established *Maximum On-Ground (MOG)* limit for a given airfield was a Port issue. This conceptual model allowed the HISA team to create a taxonomy of interface displays reflecting both (a) a logical taxonomy of subcomponents of the core referential construct (i.e., the mission), as well as (b) a reasonable categorization of known task problem features. Identification of the critical data and information necessary to portray each of these subcomponents led to the development of specialized displays (termed "Viewers") for each. One such display (the "Port Viewer") is described in more detail later.

This initial task ontology development set the stage for meeting our design goals of prioritizing "on task" user engagement. More specifically, this effort allowed us to satisfy our design criteria of maximizing explicit reference to task domain elements; maximizing cross-reference among HISA information displays with respect to core task domain concepts; and minimizing cognitive complexity in terms of interpretational demands.

IV.B. Work-Oriented Interface Implementation: The Port Viewer

The best-received of our work-oriented displays is the 'Port Viewer' illustrated below in Figure 2. The Port Viewer is a discrete interface element portraying the arrival and departure of flights for a given airfield for a given 24-hour period. This affords direct graphical summarization of conditions which planners must currently infer from a large text printout. By portraying the on-ground circumstances in one way at one time, we can allow agents to infer and depict problematical conditions (e.g., red highlighting of the period during which too many aircraft are present). In addition to displaying mission-critical information, the Port Viewer provides ready 'drill-down' capabilities via the buttons arrayed to either side of the central display. This allows planners to access additional information (e.g., airfield restrictions, clearance requirements, full data on any mission selected) without having to call up another interface unit to execute additional queries against one or more databases.

The data necessary to achieve this concise overview is currently distributed in numerous record fields among multiple databases. Some of the data required to 'draw a picture' for the user is not stored in accordance with the user's 'semantics' at all, and must be interpreted through

The Smart Lieutenant palette provides the mission planner with a single display from which he/she can access all other relevant classes of display elements. Records of missions (either specific ones or all missions for this planner) can be invoked. Alerts generated by

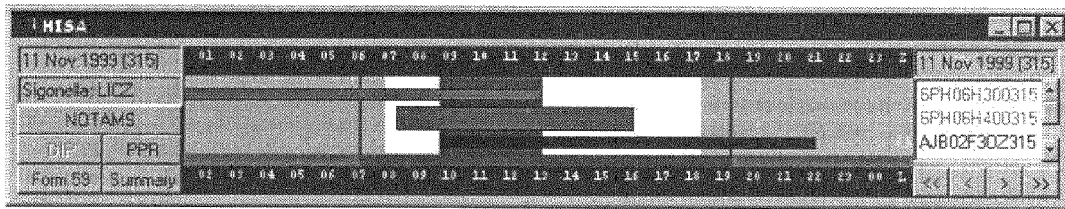


Figure 2: The Port Viewer

inference. The invocation of ASL layer agents allows for this necessary data access, fusion, and interpretation out of sight of the end user. This avoids confusing the user with unnecessary details in the data stream itself, as well as relieving the user of cognitive burdens associated with manipulating the mechanics of the database and/or making sense of the data received.

To summarize, the Port Viewer provides a unified, fused data display configured to reflect the user's task ontology, absent superfluous details and interpretational cognitive burdens which increase the potential for errors. It affords the user referential homogeneity (simplicity) with respect to data sources of high heterogeneity. It accomplishes this by according agents autonomy to perform the requisite data retrieval and fusion. In addition to satisfying the design criteria listed above for the general task ontology development, the Port Viewer illustrates minimum procedural costs for accessing and retrieving relevant data as well as maximum effective fusion of data from multiple sources.

The Port Viewer concept has received positive feedback and acceptance from the planning personnel to whom it has been demonstrated. The key to this 'payoff' has been our ability to offer HCI layer elements consistent with the ontology of the user's work and not constrained by the ontology of the supporting system(s).

IV.C.. Homogeneity of User Work Engagement: The Smart Lieutenant Palette

The most striking characteristic of channel mission planners' information systems support was its extreme heterogeneity. There was no single 'entry point' into the complexities of the mission planning, problem identification, replanning, and mission execution tasks. Our HISA interface architecture provided such an integrated entry point via the 'Smart Lieutenant' palette illustrated in Figure 3 below.

intelligent ASL agents can be managed through invocation of a pending alert queue or a historical listing of past alert conditions. Indicators on the palette cue the planner to the presence of pending alerts, as well as the arrival of new alerts since he/she last reviewed the alert queue. In a similar fashion, the planner is allowed to manage the stream of incoming queries and reports (automated stock queries), as well as to invoke a Query Assistant to generate new queries. Finally, a set of tool options allow the planner to inspect and/or manipulate agents, contacts, and preferences.

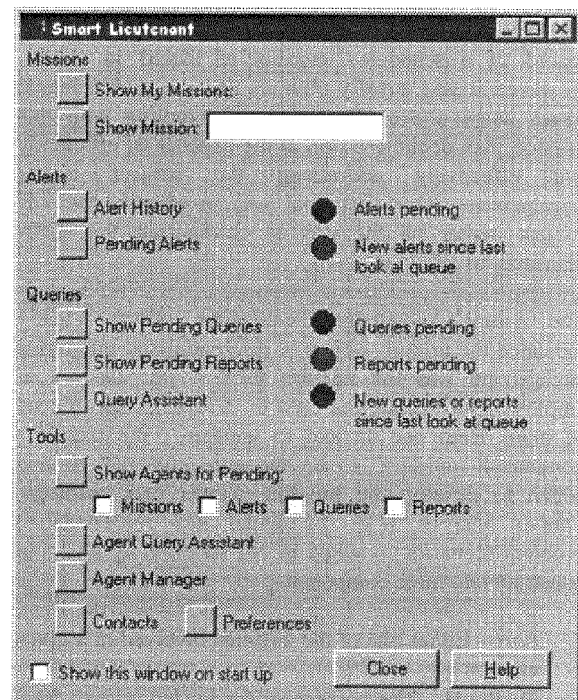


Figure 3: The Smart Lieutenant Palette

The Smart Lieutenant palette enforces referential and procedural homogeneity in the user's engagement with the obvious heterogeneity of his/her support agents and the mission planning information systems. As the top-level procedural portal, this interface element is the one most reflective of system semantics (e.g., queries, agents). However, this invocation of system ontology is. The Smart Lieutenant palette reflects our design criterion of maximum cross-reference among HISA elements with respect to the task domain constructs of (e.g.) information requests, pending issues, and workload management parameters. It maximizes effective fusion of procedural data into one concise portal for subsequent drill-down. It minimizes procedural costs by affording direct drill-down to key task elements. It minimizes cognitive complexity by summarizing the existence of alerts, queries, reports, and relevant agents.

V. Conclusion

We began this paper by discussing the command and control system of the future—the JBI—and some of the challenges and opportunities it affords HCI designers. We then discussed the role that interface agents will play in creating an environment that enables a user to remain “on task” longer, and concretized the discussion by providing example interfaces from the HISA effort. In this final section, we discuss the challenges of developing direct manipulation, work centered, interfaces for a full vision JBI. Up to this point, we have characterized the JBI mainly in terms of three different services layers. Our discussion has concentrated on an agent-based direct manipulation interface concept in the context of a distributed network-centric architecture focused on airlift command and control. It is important to recognize, however, that the full vision of a JBI involves a diverse collection of network centric systems that integrate air and space operations. One should ask if the agent-based work centered interface concept can scale to meet information usability needs for a full-blown JBI.

The JBI Information Technology concept consists of a core network system designed from the perspective of supporting battle management activities within an Air Operations Center framework. The principal goal is to enable the Joint Force Component Commander and supporting staff to make well-informed decisions that can be executed rapidly in a highly coordinated manner. Space operations, airlift, logistics, intelligence, and network security are all elements that support contingency operations. Each of these areas of the military organization are represented both in the core JBI system and via links to the extensive information networks maintain in each separate area. The essential idea for the JBI is that a core information/command and control system will be operational to support the commander within hours after approval for a contingency operation. Multiple JBIs may be in commission and operated

itself constrained with respect to task-specific features. The alert queue provides a ‘to-do list’ by which the planner can organize his/her daily itinerary. Similarly, the query features are offered with respect to the planner's work flow and activity history, and agents can be called up based upon their participation in a specific task event (mission display, alert notification). simultaneously, each sharing a common web with links to information and other resources provided by the same support agencies. In some sense, each JBI will be a “virtual” organization pulling on assets from every available source regardless of its physical location.

Clearly the level of information management associated with the JBI concept is unprecedented for military operations. Rather than reducing the “fog of war” it could in fact equally as well contribute to it. In order to insure that the JBI achieves its goal as a work support system, we believe the user interfaces each member of the JBI staff must also be regarded as a support system that is organized in a manner that keeps the worker maximally “on-task” even as the characteristics of the work problem changes based on prevailing conditions. The agent-based direct manipulation interface attempts to achieve this goal by insuring the visible portion of the interface follows a stable and consistent, yet flexible, work-oriented ontology that can dynamically connect to any appropriate information source through an interface agent that mediates ontologically differences with delivery agents. The homogeneous work centric interface focus is maintained even as the user finds the need to drill down for more detail or drill in to inspect and evaluate vital aspects of information sources, which results in dynamic connections to a pool of heterogeneous server agents, data sources, and application tools.

It should be clear that on conceptual grounds our interface concept scales to the larger arena of full battlespace management. However, on a practical levels the design task may be more challenging. One issue revolves around the semantic mapping from an information/application tool domain to the work centric one of the user. The range of information types and tools will become larger and more diverse. Can effective semantic maps be found for all of them? Clearly it would be desirable if we could establish and validate semantic mapping principles that could be used to accomplish this task. A related issue deals with the extent of automation present in the software interface mediators. In order to achieve the desired semantic mapping, interface agents may have to take on more functions that will be opaque to the user. This increases the likelihood of miscommunication of the interface to the worker—the problem of automation surprise (Woods, Sarter, and Billings, 1997). Can this problem be avoided? More research may have to be directed in this area.

To date, we have completed a preliminary demonstration of our agent-based direct manipulation concept. Initial reaction has been very favorable, and a second demonstration is scheduled. However, to more thoroughly evaluate the concept, an experiment is needed that as a minimum measures the predicted on-task/off-task time advantage and correlates it with a mission performance metric. Further, additional research will be needed to address the implications for maintaining a work centric interface focus as the properties of the interface itself expand to include such things as multi-media, multi-modal, and adaptive characteristics. Can these properties be enfolded into the agent-based direct manipulation concept? What impact might they have on semantic mapping?

Our agent-based direct manipulation interface is the first attempt we are aware of to propose a concept for how to design a collected set of work centric interfaces to a heterogeneous information network. It goes beyond the issue of standard "look and feel" that dominates user interface design today. While it may not be the final answer, we believe it is at least a useful first step to enable the JBI vision from the perspective of each individual user

References

- Bradshaw, J. M., Deutfield, S., Benoit, P., & Woolley, J. D. 1997. KaoS: Toward an Industrial Strength Open Agent Architecture. In J. M. Bradshaw (Ed.) *Software Agents*. Menlo Park, CA: AAAI Press.
- Chechile, R. A., Eggleston, R. G., Fleischman, R. N., & Sasseville, A. M. 1989. *Modeling the Cognitive Content of Displays*. Human Factors, 31(1), 31-43.
- Eggleston, R.G. 1993. *Cognitive Interface Considerations for Intelligent Cockpits*. AGARD Conference Proceedings, AGARD-CP-520.
- Finn, T., Labrou, Y., & Mayfield, J. 1997. KQML as an Agent Communication Language. In J. M. Bradshaw (Ed.) *Software Agents*. Menlo Park, CA: AAAI Press.
- Milewski, A. M., & Lewis, S. H. 1997. *Delegating to Software Agents*. International Journal of Human-Computer Studies, 46, 485-500.
- Woods, D.D., Sarter, N., & Billings, C. 1997. *Automation Surprise*. In Salvendy (Ed) Handbook of Human Factors and Ergonomics, Second Edition. New York: John Wiley & Sons, Inc.